NVM Express Technical Errata

Errata ID	**032**
Change Date	**8/30/2012**
Affected Spec Ver.	**NVM Express 1.0c**
Corrected Spec Ver.	

Submission info

Name	Company	Date
John Carroll	Intel	8/22/2012
Mikyeong Kang	Samsung	8/22/2012
Peter Onufryk	IDT	8/22/2012
Olivier Mallinger	IP Maker	8/23/2012
Matthew Wilcox	Intel	8/23/2012

Typos corrected in the specification. Device changed to controller in various places for multi-path support.

Description of the specification technical flaw:

Modify Figure 50 as shown below:

Figure 50: Firmware Image Download – Command Dword 11

Bit	Description
31:00	**Offset (OFST):** This field ~~indicates~~ specifies a 32-bit Dword offset of the firmware portion being downloaded to the controller. The offset is used to construct the complete firmware image when the firmware is downloaded in multiple pieces. The piece corresponding to the start of the firmware image shall have an Offset of 0h.

Modify a portion of section 8.1 as shown below:

If a D3 cold condition occurs during the firmware ~~update~~ activation process, the device may resume operation with either the old or new firmware.

Modify Figure 67 as shown below:

Figure 67: Identify – Power State Descriptor Data Structure

Bits	Description
103:101	Reserved
100:96	**Relative Read Throughput (RRT):** This field indicates the relative read throughput associated with this power state. The value in this field shall be less than the number of supported power states (e.g., if the controller supports 16 power states, then valid values are 0 through 15). A lower value means higher read throughput.
95:64	**Exit Latency (EXLAT):** This field indicates the maximum exit latency in microseconds associated with ~~entering~~ exiting this power state.
63:32	**Entry Latency (ENLAT):** This field indicates the maximum entry latency in microseconds associated with entering this power state.

Modify a portion of section 1.5 as shown below:

HwInit The default state is dependent on device and system configuration. The value is initialized at reset, ~~either~~ for example by an expansion ROM, or in the case of integrated devices, by a platform BIOS.

Modify a portion of section 2.1.2 as shown below:

Bits	Type	Reset	Description
08	RW / RO	0	**SERR# Enable (SEE):** Controls error reporting.~~Not supported by NVM Express.~~

Modify Figure 59 as shown below:

Figure 59: Get Log Page – Error Information Log Entry (Log Identifier 01h)

Bytes	Description
07:00	**Error Count:** This is a 64-bit incrementing error count, indicating a unique identifier for this error. The error count starts at 1h, is incremented for each unique error log entry, and is retained across power off conditions. A value of 0h indicates an invalid entry; this value may be used when there are lost entries or when there are fewer errors than the maximum number of entries the controller supports.

Modify a portion of section 5.12 as shown below:

The Set Features command uses the PRP Entry 1, PRP Entry 2, Command Dword 10, and Command Dword 11 fields. All other command specific fields are reserved.

Modify a portion of section 5.9 as shown below:

The Get Features command uses the PRP Entry 1, PRP Entry 2, ~~and~~ Command Dword 10, and Command Dword 11 fields. All other command specific fields are reserved.

Modify a portion of section 5.6 as shown below:

The command causes all commands issued to the indicated Submission Queue that are still in progress to be aborted. The controller may ~~to~~ post individual completion status of Command Aborted due to SQ Deletion for commands that have been aborted.

Modify a portion of section 1.9 as shown below:

Figure 3 illustrates the relationship between bytes, words and Dwords. A ~~q~~Qword (quadruple word) is a unit of data that is four times the size of a word; it is not illustrated due to space constraints. This specification specifies data in a little endian format.

Modify a portion of section 7.5.1 to a constant font size (9pt) as shown below:

Status of IS Register	Pin-based Action	MSI Action
One or more bits set to '1', some (but not all) bits in the IS register are cleared (i.e., host software acknowledges some of the associated completion queue entries)	Wire active	New message sent

Modify a portion of section 7.3.1 as shown below:

- ~~All controller registers defined in section 3 and internal controller state are reset.~~ The Admin Queue registers (AQA, ASQ, or ACQ) are not reset as part of a controller reset. All other controller registers defined in section 3 and internal controller state are reset.

Modify a portion of section 1.4 as shown below:

- Status for ~~device~~ controller failures (command status is processed via CQ directly)

Modify a portion of section 1.4 as shown below:

An Admin Submission and associated Completion Queue exist for the purpose of ~~device~~ controller management and control (e.g., creation and deletion of I/O Submission and Completion Queues, aborting commands, etc.) Only commands that are part of the Admin Command Set may be issued to the Admin Submission Queue.

Modify a portion of section 2.4 as shown below:

Note: It is recommended that the ~~device~~ controller allocate a unique MSI-X vector for each Completion Queue.

Modify a portion of section 6 as shown below:

The device is comprised of some number of controllers, where each controller is comprised of some number of namespaces, where each namespace is comprised of some number of logical blocks.

Modify a portion of section 6.5 as shown below:

The Compare command reads the logical blocks specified by the command from the medium and compares the data read to a comparison data buffer transferred as part of the command. If the data read from the ~~device~~ controller and the comparison data buffer are equivalent with no miscompares, then the command completes successfully. If there is any miscompare, the command completes with error. If metadata is provided, then a comparison is also performed for the metadata.

The fields used are Metadata Pointer, PRP Entry 1, PRP Entry 2, Command Dword 10, Command Dword 11, Command Dword 12, Command Dword 14, and Command Dword 15 ~~fields~~. All other command specific fields are reserved.

Modify a portion of section 8.1 as shown below:

If a D3 cold condition occurs during the firmware update process, the ~~device~~ controller may resume operation with either the old or new firmware.

Modify a portion of Figure 107 as shown below:

Bit	Description
31:00	**Expected Initial Logical Block Reference Tag (EILBRT):** This field specifies the Initial Logical Block Reference Tag expected value. This field is ~~ignored~~ only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.2.

Modify a portion of Figure 108 as shown below:

Bit	Description
31:16	**Expected Logical Block Application Tag Mask (ELBATM):** This field specifies the Application Tag Mask expected value. This field is ~~ignored~~ only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.
15:00	**Expected Logical Block Application Tag (ELBAT):** This field ~~indicates~~ specifies the Application Tag expected value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.

Modify a portion of Figure 117 as shown below:

Bit	Description
63:00	**Metadata Pointer (MD):** This field contains the address of a contiguous physical buffer of metadata, if applicable. This field shall be Dword aligned.

Modify a portion of Figure 120 as shown below:

Bit	Description
63:00	**Starting LBA (SLBA):** This field indicates the 64-bit address of the first ~~LBA~~ logical block to be read as part of the operation. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63:~~-~~32.

Modify a portion of Figure 123 as shown below:

Bit	Description
31:00	**Expected Initial Logical Block Reference Tag (EILBRT):** This field ~~indicates~~ specifies the Initial Logical Block Reference Tag expected value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.

Modify a portion of Figure 124 as shown below:

Bit	Description
31:16	**Expected Logical Block Application Tag Mask (ELBATM):** This field ~~indicates~~ specifies the Application Tag Mask expected value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.
15:00	**Expected Logical Block Application Tag (ELBAT):** This field ~~indicates~~ specifies the Application Tag expected value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.

Modify a portion of Figure 126 as shown below:

Bit	Description
63:00	**Metadata Pointer (MD):** This field contains the address of a contiguous physical buffer of metadata, if applicable. This field shall be Dword aligned.

Modify a portion of Figure 129 as shown below:

Bit	Description
63:00	**Starting LBA (SLBA):** This field indicates the 64-bit address of the first ~~LBA~~ logical block to be written as part of the operation. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63:~~-~~32.

Modify a portion of Figure 132 as shown below:

Bit	Description
31:00	**Initial Logical Block Reference Tag (ILBRT):** This field ~~indicates~~ specifies the Initial Logical Block Reference Tag value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.3.

Modify a portion of Figure 133 as shown below:

Bit	Description
31:16	**Logical Block Application Tag Mask (LBATM):** This field ~~indicates~~ specifies the Application Tag Mask value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.2.

15:00	**Logical Block Application Tag (LBAT):** This field ~~indicates~~ specifies the Application Tag value. This field is only used if the namespace is formatted to use end-to-end protection information. Refer to section 8.2.

Modify a portion of section 5.2 as shown below:

Asynchronous events are used to notify host software of error and health information as these events occur. To enable asynchronous events to be reported by the controller, host software needs to issue one or more Asynchronous Event Request commands to the controller. The controller ~~indicates~~ specifies an event to the host by completing an Asynchronous Event Request command. Host software should expect that the controller may not execute the command immediately; the command should be completed when there is an event to be reported.

Modify a portion of Figure 66 to have a constant font size (9pt) as shown below:

Bytes	O/M	Description
Controller Capabilities and Features		
77	M	**Maximum Data Transfer Size (MDTS):** This field indicates the maximum data transfer size between the host and the controller. The host should not submit a command that exceeds this transfer size. If a command is processed that exceeds the transfer size, then the command is aborted with a status of Invalid Field in Command. The value is in units of the minimum memory page size (CAP.MPSMIN) and is reported as a power of two (2^*n*). A value of 0h indicates no restrictions on transfer size. The restriction includes metadata if it is interleaved with the logical block data.

Disposition log

8/22/2012	Erratum captured.
8/28/2012	Added command Dword 11 to get features, additional typos corrected, changed instances of device to controller and indicates to specifies
8/29/2012	Added context to tables & fixed font sizes
8/30/2012	Clarified SERR# Enable (SEE) RW requirement.
10/2/2012	Erratum ratified.

Technical input submitted to the NVMHCI Workgroup is subject to the terms of the NVMHCI Contributor's agreement.